

### Amendments to the Claims

1 Claim 1 (currently amended): A method of ~~improving~~ data transfer in a virtual server  
2 environment of a computing network, the method comprising ~~steps of~~:

3 receiving ~~one or more~~ a plurality of packets to be routed to or from a plurality of virtual  
4 servers operating in a single physical device;

5 providing an internal routing table for data link layer routing to or from selected ones of  
6 the virtual servers, wherein entries in the internal routing table are learned dynamically while  
7 processing selected ones of the received packets at a network layer; and

8 using the internal routing table for routing other ones of the received packets to or from  
9 the selected ones of the virtual servers at the data link layer,

10 wherein:

11 the selected ones of the received packets comprise, for each supported pair of  
12 input data link layer component and output data link layer component, a first-processed one of the  
13 packets which arrives using the input data link layer component and which is addressed to the  
14 output data link layer component; and

15 the other ones of the received packets comprise, for each of the supported pairs of  
16 input data link layer component and output data link layer component, subsequently-processed  
17 ones of the packets which arrive using the input data link layer component and which are  
18 addressed to the output data link layer component.

1 Claim 2 (currently amended): A method of ~~improving~~ data transfer in a virtual server  
2 environment of a communications network, the method comprising ~~steps of~~:

3 providing a concentrator that combines traffic from a plurality of virtual servers operating  
4 in a single physical device into a single outbound stream; and  
5 routing packets of the combined traffic, further comprising ~~steps of~~:  
6 intercepting packets of the traffic at a data link layer of a communications protocol  
7 stack;  
8 comparing a destination address of each intercepted packet to entries in a data link  
9 layer routing table comprising at least one entry, each entry specifying an input data link layer  
10 component, output data link layer component pair, to determine if a matching entry is present in  
11 the table, the matching entry specifying a data link layer component on which the intercepted  
12 packet arrived as the input data link layer component of the pair and the destination address of the  
13 intercepted packet as the output data link layer component of the pair;  
14 forwarding the intercepted packet to a higher layer of the communications  
15 protocol stack if ~~[[no]] the matching entry is not found by the comparing step~~, for routing by the  
16 higher layer; and  
17 performing data link layer routing of the intercepted packet, without intervention  
18 of the higher layer, if ~~[[a]] the matching entry is found by the comparing step~~.

1 Claim 3 (currently amended): The method according to Claim 2, wherein ~~the step of~~ performing  
2 data link layer routing further comprises ~~steps of~~:

3 replacing ~~[[the]]~~ an inbound packet header of the intercepted packet with an outbound  
4 packet header using information from the matching entry, thereby creating a modified packet  
5 header; and

6 forwarding the intercepted packet using the modified packet header.

1 Claim 4 (currently amended): The method according to Claim 2, wherein the entries in the data  
2 link layer routing table are dynamically learned and further comprising:

3 adding a new entry to the table for each of the intercepted packets for which the matching  
4 entry is not found and for which the data link layer component on which the intercepted packet  
5 arrived and the output data link layer component that matches the destination address of the  
6 intercepted packet are both supported, the new entry specifying the data link layer component on  
7 which the intercepted packet arrived as the input data link layer component of the pair and the  
8 output data link layer component that matches the destination address of the intercepted packet as  
9 the output data link layer component of the pair.

1 Claim 5 (currently amended): The method according to Claim 2, wherein at least one ~~or more~~ of  
2 the virtual servers is an ~~[[are]] application servers~~ server.

1 Claim 6 (currently amended): The method according to Claim 2, wherein the virtual servers each  
2 operate in a logical partition within ~~[[a]]~~ the single physical ~~computing~~ device.

1 Claim 7 (currently amended): The method according to Claim 2, further comprising ~~the step of~~  
2 deleting selected entries from the data link layer routing table when the selected entries become  
3 obsolete.

1 Claim 8 (currently amended): A system for ~~improving~~ data transfer in a virtual server  
2 environment of a communications network, comprising:

3 means for providing a concentrator that combines traffic from a plurality of virtual servers  
4 operating in a single physical device into a single outbound stream; and

5 means for routing packets of the combined traffic, further comprising:

6 means for intercepting packets of the traffic at a data link layer of a  
7 communications protocol stack;

8 means for comparing a destination address of each intercepted packet to entries in  
9 a data link layer routing table comprising at least one entry, each entry specifying an input data  
10 link layer component, output data link layer component pair, to determine if a matching entry is  
11 present in the table, the matching entry specifying a data link layer component on which the  
12 intercepted packet arrived as the input data link layer component of the pair and the destination  
13 address of the intercepted packet as the output data link layer component of the pair;

14 means for forwarding the intercepted packet to a higher layer of the  
15 communications protocol stack if ~~[[no]]~~ the matching entry is not found by the means for  
16 comparing, for routing by the higher layer; and

17 means for performing data link layer routing of the intercepted packet, without  
18 intervention of the higher layer, if ~~[[a]]~~ the matching entry is found by the means for comparing.

1 Claim 9 (currently amended): The system according to Claim 8, wherein the means for  
2 performing data link layer routing further comprises:

3 means for replacing ~~[[the]]~~ an inbound packet header of the intercepted packet with an

4 outbound packet header using information from the matching entry, thereby creating a modified  
5 packet header; and

6 means for forwarding the intercepted packet using the modified packet header.

1 Claim 10 (currently amended): The system according to Claim 8, wherein the entries in the data  
2 link layer routing table are dynamically learned, the dynamic learning further comprising:

3 means for adding a new entry to the table for each of the intercepted packets for which the  
4 means for comparing does not find the matching entry and for which the data link layer  
5 component on which the intercepted packet arrived and the output data link layer component that  
6 matches the destination address of the intercepted packet are both supported, the new entry  
7 specifying the data link layer component on which the intercepted packet arrived as the input data  
8 link layer component of the pair and the output data link layer component that matches the  
9 destination address of the intercepted packet as the output data link layer component of the pair.

1 Claim 11 (currently amended): The system according to Claim 8, wherein at least one ~~or more~~ of  
2 the virtual servers ~~[[are]]~~ is an application servers server.

1 Claim 12 (currently amended): A computer program product for ~~improving~~ data transfer in a  
2 virtual server environment of a communications network, the computer program product having  
3 computer usable code embodied on one or more computer readable media and therewith, the  
4 computer program product comprising:

5 computer readable program code ~~[[means]]~~ for providing a concentrator that combines

6 traffic from a plurality of virtual servers operating in a single physical device into a single  
7 outbound stream; and

8 computer readable program code [[means]] for routing packets of the combined traffic,  
9 further comprising:

10 computer readable program code [[means]] for intercepting packets of the traffic  
11 at a data link layer of a communications protocol stack;

12 computer readable program code [[means]] for comparing a destination address of  
13 each intercepted packet to entries in a data link layer routing table comprising at least one entry,  
14 each entry specifying an input data link layer component, output data link layer component pair,  
15 to determine if a matching entry is present in the table, the matching entry specifying a data link  
16 layer component on which the intercepted packet arrived as the input data link layer component of  
17 the pair and the destination address of the intercepted packet as the output data link layer  
18 component of the pair;

19 computer readable program code [[means]] for forwarding the intercepted packet  
20 to a higher layer of the communications protocol stack if [[no]] the matching entry is not found by  
21 the computer readable program code [[means]] for comparing, for routing by the higher layer; and

22 computer readable program code [[means]] for performing data link layer routing  
23 of the intercepted packet, without intervention of the higher layer, if [[a]] the matching entry is  
24 found by the computer readable program code [[means]] for comparing.

1 Claim 13 (currently amended): The computer program product according to Claim 12, wherein  
2 the computer readable program code [[means]] for performing data link layer routing further

3 comprises:

4 computer readable program code ~~[[means]]~~ for replacing ~~[[the]]~~ an inbound packet header  
5 of the intercepted packet with an outbound packet header using information from the matching  
6 entry, thereby creating a modified packet header; and

7 computer readable program code ~~[[means]]~~ for forwarding the intercepted packet using  
8 the modified packet header.

1 Claim 14 (currently amended): The computer program product according to Claim 12, wherein  
2 the entries in the data link layer routing table are dynamically learned, the dynamic learning further  
3 comprising:

4 computer readable program code for adding a new entry to the table for each of the  
5 intercepted packets for which the computer readable program code for comparing does not find  
6 the matching entry and for which the data link layer component on which the intercepted packet  
7 arrived and the output data link layer component that matches the destination address of the  
8 intercepted packet are both supported, the new entry specifying the data link layer component on  
9 which the intercepted packet arrived as the input data link layer component of the pair and the  
10 output data link layer component that matches the destination address of the intercepted packet as  
11 the output data link layer component of the pair.

1 Claim 15 (currently amended): The computer program product according to Claim 12, wherein at  
2 least one ~~or more~~ of the virtual servers ~~[[are]]~~ is an application ~~servers~~ server.